複数の秘密を考慮したパスワード認証付き秘密分散法

複数の秘密を考慮したパスワード認証付き秘密分散法
A Password-Protected Secret Sharing Supporting Multiple Secrets

新井　貴大
Takahiro Arai
法政大学大学院情報科学研究科情報科学専攻
E-mail: takahiro.arai.2v@stu.hosei.ac.jp

Abstract

Password-Protected Secret Sharing (PPSS) presented by Bagherzandi et al. is proposed in order to resolve drawback of secret sharing which is unauthorized users can access storages storing partial information can reconstruct a secret. PPSS is a secret sharing that ensures only the owner of the secret who knows correct password to obtain the original secret by applying password authentication to partial information. But, their model requires secure channel between user and servers and independent secret/public key pair at the distribution phase for each secret. When a secret is large, their scheme encrypts the secret with symmetric key encryption (SKE) and the symmetric key with CPA secure public key encryption (PKE). Because of such combination, it seems difficult to prove strong security (i.e., CCA security) of their scheme at least in the standard model. In this paper, we propose a new PPSS model and scheme. Proposed model deals with multiple secrets with using a single secret key/public key pair and does not require secure channel during the distribution phase. Proposed scheme does not use a simple combination of SKE and PKE but use Kurosawa-Desmedt hybrid encryption that is proven to be CCA secure in the standard model, and is constructed by combining public key encryption part of this hybrid encryption with password authentication. The scheme is expected to be more secure than that of Bagherzandi et al.

1. 研究背景

近年, DropBox や Google Drive 等のクラウドストレージが多く利用されている．しかし，ほとんど全てのクラウドストレージでは保存データが暗号化されず，保存データが漏洩してしまう可能性が考えられる．

データ漏洩に対し，2 つの対策技術が挙げられる．一方はデータを暗号化方式によって暗号化した後に保存する方法であり，もう一方は秘密分散法を用いて保存したいデータの部分情報を複数のサーバに保存する方法である．秘密分散法とは秘密情報をシェアと呼ばれる複数の部分情報として分散し，一定数以上のシェアが集まらない限り元の秘密情報の復元ができない技術である．特に，(k, n) しきい値秘密分散法 [6] では n 個のシェアのうち，少なくとも k 個のシェアがなければ復元することができない．

一方，暗号化方式を利用した場合，復号鍵の安全な管理が問題となる．また，秘密分散を利用した場合，安全性がサーバの安全性に依存する問題がある．例えば，サーバ管理者等の不正により，復元が行われる可能性がある．そのため，どちらの場合も情報漏洩の危険を無くすことができない．

これらの対策として，秘密分散のシェアにパスワード認証を組み合わせることで，復元時に認証を行うパスワード認証付き秘密分散法 (Password-Protected Secret Sharing: PPSS) が Bagherzandi らによって提案された [3]．この方式は，秘密とパスワードを準同型性を持つ公開鍵暗号である ElGamal 暗号 [4] で暗号化し，その秘密鍵を秘密分散で分散し，ElGamal 暗号の準同型性を用いてパスワード認証を行う．また，$k-1$ 台までのサーバの結託に対して安全である．

この方式は，秘密のサイズが 2048bit 以上の場合，秘密を共通鍵暗号で暗号化し，共通鍵暗号の鍵を公開鍵暗号で暗号化する．この組み合わせは，仮に安全性が証明できたとしても CCA 安全といった強い安全性を持たず，妥当性が不明確なランダムオラクルが必要となる．また，悪意のある攻撃者によって通信路が盗聴された場合，通信路上の秘密の暗号文と全ての秘密鍵のシェアから元の秘密が攻撃者によって復元されるため，秘密を分散する度にユーザ・サーバ間の安全な通信路が必ず必要となる問題がある．

そこで本研究では，これらの欠点を改善した PPSS の新しいモデルと新しい方式の提案を行う．新しいモデルとして，単一の秘密鍵・公開鍵のみを使用する場合でも，複数の秘密を扱うことができ，分散時に安全な通信路を必要としないモデルを提案する．新しい方式として，共通鍵暗号と ElGamal 暗号の単純な組み合わせではなく，スタンダードモデルで CCA 安全であることが証明されている Kurosawa-Desmedt ハイブリッド暗号 [5] を使用することで，Bagherzandi らの方式 [3] より安全であることが期待される方式を提案モデル上で提案する．

2. 準備

本節では本論文で提案方式を構成する暗号技術について説明する．

2.1. (k, n) しきい値秘密分散法 [6]

秘密分散法は暗号技術の 1 つであり，Shamir によって

Superviser: Prof. Satoshi Obana

提案された (k,n) しきい値秘密分散法 [6] が最も有名な方式である．n 個のシェアのうち最低 k 個から復元が行うことができる (k,n) しきい値秘密分散法は以下の 2 つ組アルゴリズム $(\mathsf{Dist},\mathsf{Rec})$ からなる．

分散 $(Dist)$: 秘密を $s \in \mathbb{F}$ として $f(0)=s$ となる $\mathbb{F}$ 上の $k-1$ 次の多項式 $f(x)$ をランダムに選び，サーバ $i \in \{1,\ldots,n\}$ に $s^{(i)}=f(i)$ をシェアとして送信する．

復元 (Rec): $k'(\geq k)$ 台のサーバからそれぞれのシェア $(s^{(i_1)},\ldots,s^{(i_{k'})})$ を得た時，全ての $j \in \{1,\ldots,k'\}$ に対して，$f(i_j)=s^{(i_j)}$ を満たす $k-1$ 次の多項式 $f(x)$ が一意に定まり，$s=f(0)$ を秘密として復元，出力する．

2.2. ElGamal 暗号 [4]

準同型性を持つ公開鍵暗号である ElGamal 暗号 [4] は以下の 3 つ組アルゴリズム $(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ からなる．

鍵生成 (Gen): 巡回群 G において，位数素数が q であり，$|q|=\lambda$bit(λ:セキュリティパラメータ) であるものを選び，G の生成元 g を選択する，x を $\mathbb{Z}_q=\{0,\ldots,q-1\}$ からランダムに選択し，$y=g^x$ を計算し，公開鍵 $pk=(G,q,g,y)$，秘密鍵 $sk=x$ とする．

暗号化 (Enc): $m \in G$ を平文とし，乱数 r を $\mathbb{Z}_q$ からランダムに選び，公開鍵 pk から $(c_1,c_2)=(g^r, m\cdot y^r)$ を計算し，暗号文ペアとする．

復号 (Dec): 暗号文ペア (c_1,c_2) と秘密鍵 x から $c_2\cdot(c_1^x)^{-1}=m'$ を計算し，復号文する．

ElGamal 暗号は次の準同型性を持つ．ある平文 m,m' に対する暗号文を $(c_1,c_2)=(g^r,m\cdot y^r),(c_1',c_2')=(g^{r'},m'\cdot y^{r'})$ とした時，$(c_1\cdot c_1', c_2\cdot c_2')$ を計算することで，$m\cdot m'$(m と m' を乗算した値) の暗号文を計算することができる．また，$g^m, g^{m'}$ に対する暗号文を $(c_1,c_2)=(g^r,g^m\cdot y^r),(c_1',c_2')=(g^{r'},g^{m'}\cdot y^{r'})$ とした時，$(c_1\cdot c_1', c_2\cdot c_2')$ を計算することで，$g^{m+m'}$ の暗号文を計算することができる．よって，$m+m'$ が小さい場合，$g^{m+m'}$ から $m+m'$ を計算することができ，この形の ElGamal 暗号を modified-ElGamal 暗号と呼ぶ．

ElGamal 暗号は，DDH 仮定のもとでターゲットとする暗号文を受け取る前後に，自分で選んだ平文に対する暗号文を得ることができる選択平文攻撃 (Chosen Plaintext Attack:CPA) を行う攻撃者に対して，暗号文から平文のいかなる情報も漏れない識別不可能性 (indistinguishability:IND) を満たす IND-CPA 安全であることが証明されている．

2.3. Kurosawa-Desmedt 暗号 [5]

公開鍵暗号と共通鍵暗号を組み合わせたハイブリッド暗号の 1 つである Kurosawa-Desmedt 暗号 [5] は以下の 3 つ組アルゴリズム $(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ からなる．

鍵生成 (Gen): G を位数素数 q の可換群，H を $G\rightarrow\{0,1\}^k$ となるハッシュ関数，TCR を κ をインデックスとする 2 つ以上のターゲットが衝突しないことを保証するハッシュ関数とする．そして，生成元 g_1,g_2 を G から，x_1,x_2,y_1,y_2 を $\mathbb{Z}_q$ からランダムに選び，$c=g_1^{x_1}g_2^{x_2}, d=g_1^{y_1}g_2^{y_2}$ を計算し，(x_1,x_2,y_1,y_2) を秘密鍵，(g_1,g_2,c,d,κ) を公開鍵とする．秘密鍵・公開鍵は公開鍵暗号部分に相当する鍵カプセル化メカニズム (Key Encapsulation Mechanism: KEM)[7] の鍵である．

暗号化 (Enc): $m \in G$ を平文とし，r を $\mathbb{Z}_q$ からランダムに選び，以下の式を計算する．

$$u_1=g_1^r, u_2=g_2^r, \alpha=\mathsf{TCR}(\kappa;u_1,u_2),$$
$$v=c^r d^{r\alpha}, H(v)=K=(K_0,K_1)$$
$$\chi=\mathsf{DEM.Enc}(K_0,m), a=\mathsf{MAC}(K_1,\chi)$$

暗号文は (u_1,u_2,χ,a) とする．ただし，MAC はメッセージ認証のタグとし，DEM.Dec はブロック暗号を用いて m を暗号化する．共通鍵 K は共通鍵暗号部分に相当するデータカプセル化メカニズム (Deta Encapsulation Mechanism: DEM)[7] の鍵であり，(u_1,u_2) は KEM の暗号文，(χ,a) は DEM の暗号文である．

復号 (Dec): 暗号文 (u_1,u_2,χ) から以下の式を計算する．

$$\alpha=\mathsf{TCR}(\kappa;u_1,u_2),$$
$$v'=u_1^{x_1+y_1\alpha}u_2^{x_2+y_2\alpha}, H(v')=K'=(K_0',K_1')$$

$\mathsf{MAC}(K_1',\chi)=a$ が成立しなければ $reject$，成立すれば $m'=\mathsf{DEM.Dec}(K_0',\chi)$ を計算する．

Kurosawa-Desmedt 暗号は，DDH 仮定のもとで選択平文攻撃でできる攻撃に加えて，ターゲットとする暗号文を受け取る前後に，自分で選んだターゲット以外の暗号文の復号結果を返す復号オラクルに質問できる選択暗号文攻撃 (Chosen Ciphertext Attack:CCA) を行う攻撃者に対して，識別不可能性 (IND) を満たす IND-CCA 安全であることが証明されている．CCA は CPA より強力な安全性であり，IND-CPA<IND-CCA である．IND-CCA は公開鍵暗号で最も強力な安全性である．

3. 既存の PPSS[3]

本節では，Bagherzandi らによって提案された PPSS[3] とその問題点について説明する．

3.1. 既存モデル

既存の PPSS モデルは以下の 3 つ組アルゴリズム $(\mathsf{Init},\mathsf{User},\mathsf{Server})$ からなる．

初期化アルゴリズム $(\mathsf{Init}(p,s))$:設定パスワード p と秘密情報 s を入力とし，公開パラメータ st_0 とサーバ S_i の秘密状態 st_i の組 $st=(st_0,st_1,\ldots,st_n)$ を出力する．

ユーザアルゴリズム $(\mathsf{User}(\tilde{p},st_0))$:ユーザによって実行される対話型アルゴリズムで入力パスワード $\tilde{p}$ と公開パラメータ st_0 を入力とし，復元された情報 s' または，拒否を示す $\perp$ を出力する．

サーバアルゴリズム $(\mathsf{Server}(st_i))$: サーバ S_i によって実行される対話型アルゴリズムで，サーバの秘密状態 st_i を入力とし，出力はない．分散は初期化アルゴリズム，復元はユーザアルゴリズムとサーバアルゴリズムが対話することで行われる．

3.2. 既存モデルの安全性定義

既存の PPSS は，パスワードを知らない攻撃者が元の秘密を得ることができないことを要求する．正しいパスワードを知らず，公開パラメータ st_0 と $k-1$ 台

までのサーバの秘密状態 s_i を知ることができる攻撃者が正しいパスワードを入力することを試みることを想定する．この時，元の秘密を得ることができる確率が，$\frac{\text{パスワード試行回数}}{\text{パスワード辞書サイズ}}$ 未満の場合，PPSS は安全である．

3.3. 既存方式

Bagherzandi らによって提案された PPSS は 1 人のユーザ U と n 台のサーバ $\{S_j\}_{j=1}^{n}$ により実行され，以下の 2 つのフェイズ (Dist, Rec) からなる．

分散 ($Dist$): U は，ElGamal 暗号の鍵生成を行い，秘密鍵 x のシェア $x^{(j)}$ を計算し，秘密 s の ElGamal 暗号文 (c_s, d_s) とパスワード p の modified-ElGamal 暗号文 (c_p, d_p) とともにサーバ S_j に送信する．

復元 (Rec): 入力パスワード $\tilde{p}$ を入力するユーザ U は，n 台の中から選んだ $k'(\geq k)$ 台のサーバに $\tilde{p}$ の modified-ElGamal 暗号文 $(c_{\tilde{p}}, d_{\tilde{p}})$ を送信する．そして，$\{S_j\}_{j=1}^{k'}$ と U が通信・計算を行い，最終的に S_j が $s \cdot g^{(p-\tilde{p})\sum_i t_i}(t_i$ は S_i が生成する乱数) の ElGamal 暗号文を U に送信し，U はしきい値復号を行う．復号結果は $p \neq \tilde{p}$ の時，元の秘密 s となり，それ以外の時は乱数となるため，不正なユーザには元の秘密に関係する情報が漏れない．

■既存方式の問題点

この方式の問題点は，秘密が群サイズ (通常 2048bit) を超える場合は共通鍵を $H_k(r)$ とした共通鍵暗号で秘密を暗号化し，r を ElGamal 暗号で暗号化するハイブリッド暗号を用いるが，このハイブリッド暗号は安全性が証明されていないことと，仮にできても妥当性に疑問が残るランダムオラクルが必要となることである．

さらに，$k-1$ 台までのサーバと結託できる攻撃者が通信路を盗聴することを考えた場合，攻撃者が通信路上から得た秘密の暗号文と秘密鍵のシェアから元の秘密を復元することが可能である．この通信路盗聴は秘密鍵のシェアを固定して予めサーバに置くことで，防ぐことができるように思える．しかし，攻撃者は，結託サーバから得た秘密の暗号文と攻撃者が決めたパスワードの暗号文を全サーバに送信し，攻撃者が決めたパスワードを用いて復元を行うことで，正しいパスワードを知っているかどうかに関わらず，元の秘密が復元できてしまう問題がある．この問題は，モデルが複数の秘密を考慮していないことと，秘密の暗号文が CPA 安全である ElGamal 暗号で暗号化されることに起因する．

4. 複数の秘密を扱う PPSS

既存 PPSS の問題を解決すべく，複数の秘密を扱う提案モデルとその安全性を定義し，秘密を CCA 安全な暗号方式で暗号化する方式を提案し，安全性議論を行う．

4.1. 複数の秘密を扱う PPSS モデル

このモデルの主な利点は，既存モデルは単一の秘密しか扱えないことに対して，複数の秘密を扱えることであり，本モデルは既存モデルの拡張モデルである．

提案モデルは以下の 4 つ組アルゴリズム (Init, Enc, Entry, Rec) からなる．

初期化アルゴリズム (Init()):このアルゴリズムは，ユーザによって実行され，入力はなく，公開パラメータ st_0 とサーバ S_i の秘密状態 st_i の組 $st = (st_0, st_1, \ldots, st_n)$ を出力する．このアルゴリズムは，既存方式 [3] の Init の一部に相当する．

暗号化アルゴリズム (Enc(p, s, st_0)): このアルゴリズムは，ユーザによって実行され，設定パスワード p，秘密 s と公開パラメータ st_0 を入力とし，パスワードの暗号文 C_p，秘密の暗号文 C_s を出力する．このアルゴリズムは，既存方式 [3] の Init の Init の一部 (提案アルゴリズムの Init に相当しない部分) と User アルゴリズムの全体に相当する．

登録アルゴリズム (Entry(C_p, C_s)): このアルゴリズムは，サーバによって実行され，パスワードの暗号文 C_p と秘密の暗号文 C_s を入力とし，暗号文に関連付けられた暗号文識別子 cid を出力する．このアルゴリズムはサーバにパスワードと秘密の暗号文を登録する．

復元アルゴリズム (Rec$(cid, \tilde{p}, st_0, st_i)$): このアルゴリズムは，サーバによって実行され，暗号文識別子 cid，入力パスワード $\tilde{p}$，公開パラメータ st_0，サーバ S_i の秘密状態 st_i を入力とし，復元された情報 s' または，拒否を示す $\perp$ を出力する．このアルゴリズムは，既存方式 [3] の User と Server に相当する．

このモデルの流れは，最初に Init が 1 度だけ実行され，その後 Enc,Entry,Rec が任意回数実行される．ただし，Rec が実行される前に対応する Enc と Entry が実行されている必要がある．また，Init は，公開鍵暗号の Gen，Enc は公開鍵暗号の Enc に相当し，Entry と Rec は公開鍵暗号の Dec に相当する．

このモデルは，攻撃者がマルウェア等による遠隔操作でユーザの PC を自由に操作できる状況を想定する．この状況では，攻撃者は PPSS プロトコルを何度でも実行でき，サーバに送る情報を入手することができる．

任意のプロトコル PPSS = (Init, Enc, Entry, Rec) が以下の正当性を満たす時，プロトコル PPSS は提案モデル上の PPSS である．また，頑強性，健全性も満たす必要がある．

正当性 任意の秘密 s に対し，以下が圧倒的である．
$\Pr(st = (st_0, st_1, \ldots, st_n) \leftarrow \text{Init}(), \{(C_{sj}, C_{pj}) \leftarrow \text{Enc}(p_j, s_j, st_0), cid_j \leftarrow \text{Entry}(C_{sj}, C_{pj})\}_{j=1}^{m} : \{\{\text{Rec}(cid_j, p_j, st_0, st_i)\}_{i=1}^{k} = s_j\}_{j=1}^{m})$

頑強性 ユーザが通信する n 台のサーバの内，少なくとも k 台がプロトコルに従う時，元の秘密を復元できる．

健全性 例え，n 台のサーバが攻撃者と結託していても，定められた公開パラメータ st_0 を使う正規のユーザは元の秘密とは異なる秘密を復元しない．

4.2. 複数の秘密を扱う PPSS の安全性定義

提案モデルの安全性定義では，3.2 節の既存方式の安全性定義に加えて，提案モデルのアルゴリズムをオラクル (* で示す) に置き換えて，挑戦者と攻撃者とのゲームを行うことを考える．ゲームは以下の流れで行われる．

1. 挑戦者は Init* を実行し，攻撃者に公開パラメータと結託サーバの秘密状態を送る．

2. 攻撃者はターゲットの秘密に関連するような (ただし，ターゲット以外) 新しい秘密の暗号文を作成する．そして，挑戦者の Entry^* を用いてサーバに暗号文を登録し，攻撃者のパスワード入力とした Rec^* を用いてターゲットの秘密に関連するような秘密の暗号文を復元する．この操作は任意の回数行うことができる．ただし，Enc^* はターゲットの秘密の暗号化以外には用いず，それ以外の暗号化は攻撃者が独自に行うことができる．また，Entry^* と Rec^* は，単一のサーバに対応しているため，k 回以上実行する必要がある．
3. 攻撃者は，任意の秘密 (s_0, s_1) を挑戦者の Enc^* に送り，Enc^* は $b = 0$ or 1 をランダムに選んだ s_b とパスワード $p =$ 乱数として Enc^* に入力する．このとき暗号化された方の秘密と乱数の暗号文をターゲットの暗号文とし，これらの暗号文を攻撃者に送る．
4. 攻撃者は 2. と同様の操作を行う．ただし，ターゲットの秘密に対して Rec^* を用いてパスワードの入力を試みることができる．

5. 攻撃者は，ターゲットの秘密が s_0 か s_1 かを当てる．この時，攻撃者がターゲットの秘密を当てる確率が $\frac{1}{2}$ かつ，Rec^* の入力で正しいパスワードを当てる確率が $\frac{\text{パスワード試行回数}}{\text{パスワード辞書サイズ}}$ 未満の場合，その方式は安全である．つまり，このゲームを通じて攻撃者が s_0 と s_1 の暗号文を区別できず，正しいパスワードがわからない場合安全である．このゲームの操作はパスワード試行部分を除いて CCA モデルに酷似しており，提案モデルは CCA モデルに非常に近いモデルであるといえる．

4.3. 複数の秘密を扱う PPSS の方式

■方式のアイデア

提案の基本的なアイデアは，(1)ElGamal 暗号の秘密鍵シェア，(2)Kurosawa-Desmedt 暗号の秘密鍵シェア，(3) 秘密の暗号文シェア，(4) 秘密の暗号文のシェアから生成した MAC を各サーバに保存することである．また，復元時には既存方式と同様の ElGamal 暗号の準同型を用いたパスワード認証を Kurosawa-Desmedt 暗号の KEM と組み合わせ, ElGamal 暗号と Kurosawa-Desmedt 暗号のしきい値復号を同時に行う．このことにより，パスワードを知らない不正なユーザが元の秘密を得ることができないことを保証する．ただし，ElGamal 暗号と Kurosawa-Desmedt 暗号を同時にしきい値復号を行わなければパスワード認証が行えず，安全性を保つことができない．さらに，秘密の暗号文シェアは，秘密を共通鍵暗号で暗号化した後，秘密分散の順に行う必要があり，逆に行うと計算量が増加してしまう．また，MAC 値を秘密の暗号文のシェアから生成することで，既存方式と異なり，サーバ側で各サーバが独立にパスワード認証可能であり，サーバ側でアクセスをコントロールできる．

安全性の改善の本質は以下の 4 点である．(1) 秘密自体は CCA 安全である Kurosawa-Desmedt 暗号で暗号化していること，(2)KEM 部分は CCA 安全ではないが，全体が CCA 安全である Kurosawa-Desmedt 暗号と組み合わせることで全体が CCA 安全になること見込めること，(3) 初期化時の秘密鍵シェアの送信時以外には，安全な通信路が必要ないこと，(4) 各サーバで，秘密の暗号文シェアから求めた異なる MAC 値を持ち，各サーバで認証可能であるため，同じ攻撃者からの総当たり攻撃をサーバ側で防ぐことができることである．ハイブリッドバージョンの既存の方式で，提案方式のように秘密の暗号文をシェアとしてサーバに保持しても，サーバ側で認証する術がないため，このような改善はできない．

■提案方式の詳細

提案方式は 1 人のユーザ U と n 台のサーバ $\{S_j\}_{j=1}^n$ により実行され，$(\mathsf{Init}, \mathsf{Enc}, \mathsf{Rec})$ のフェイズからなる．

初期化 (Init): U は，$|q|$ と $|q'|$ が λbit (λ:セキュリティパラメータ) かつ，$q|q'-1$ となる素数 q, q' から位数が q である群 $G = \mathbb{Z}_q$ の $g^2 \neq 1$ かつ $g^{q'} = 1$ となる生成元 g_1, g_2, g_3 を選び，x_1, x_2, y_1, y_2, x_3 を $\mathbb{Z}_{q'}$ から選ぶ．また，パスワード辞書 $D = \mathbb{Z}_{q'}$ とする．ここで，$c = g_1^{x_1} g_2^{x_2}, d = g_1^{y_1} g_2^{y_2}, y_3 = g_3^{x_3}$ を計算し，TCR のインデックス κ とし，公開鍵 $(q, q', g_1, g_2, g_3, c, d, y_3, \kappa)$，秘密鍵 $(x_1, x_2, y_1, y_2, x_3)$ とする．さらに，$\{f_{x_i}(0) = x_i\}_{i=1}^3, \{f_{y_i}(0) = y_i\}_{i=1}^2$ となる係数が乱数である $k-1$ 次の多項式関数 $f_{x_1}, f_{x_2}, f_{y_1}, f_{y_2}, f_{x_3}$ を $\mathbb{Z}_{q'}$ 上でそれぞれ定義し，$\{f_{x_i}(j) = x_i^{(j)}\}_{i=1}^3, \{f_{y_i}(0) = y_i^{(j)}\}_{i=1}^2$ を秘密鍵のシェアとしてそれぞれ $\{S_j\}_{j=1}^n$ に送信する．

暗号化 (Enc): 秘密 s に対し，r を $\mathbb{Z}_{q'}$ からランダムに選び，$u_1 = g_1^r, u_2 = g_2^r, \alpha = \mathsf{TCR}(\kappa; u_1, u_2), v = c^r d^{r\alpha}$ から共通鍵 $H(v) = K = (K_0, K_1)$ を計算し，s を $\chi = \mathsf{DEM.Enc}(K_0, s)$ として暗号化する．さらに，χ を $\chi^{(i)}$ として，(k, n) しきい値秘密分散で分散し，$a_i = \mathsf{MAC}(K_1, \chi^{(i)})$ を計算する．

さらに，設定パスワードを $p \in D$，$r_p \in_R \mathbb{Z}_{q'}$ から $(c_p, d_p) = (g_3^{r_p}, y_3^{r_p} \cdot g_3^p)$ を計算する．

U は最後に $(u_1, u_2, \chi^{(i)}, a_i, c_p, d_p)$ を暗号文として各サーバ S_j に送信する．暗号文を受け取った S_j は，暗号文をサーバに登録し，暗号文識別子 cid を U に送信する．ただし，受け取った暗号文の中に既にサーバ上に同じ値がある場合は，登録せずに拒否し，U に通知する．

復元 (Rec): ユーザ U と n 台の中から選んだ $k'(\geq k)$ 台のサーバ $\{S_j\}_{j=1}^{k'}$ が以下を実行することで復元を行う．

1. U は，暗号文識別子 cid と入力パスワード $\tilde{p} \in D$ を modified-ElGamal 暗号で暗号化した $(c_{\tilde{p}}, d_{\tilde{p}}) = (g_3^{r_{\tilde{p}}}, y_3^{r_{\tilde{p}}} g_3^{\tilde{p}})$ を各 S_j に送信する．ただし，正規のユーザならば $\tilde{p} = p$ である．
2. 各 S_j は $(c_p / c_{\tilde{p}}, d_p / d_{\tilde{p}})$ と $\mathbb{Z}_{q'}$ からランダムに選んだ t_j に対して $(c_{\beta_j}, d_{\beta_j}) = (g^{(r_p - r_{\tilde{p}})t_j}, y^{(r_p - r_{\tilde{p}})t_j} \cdot g^{(p-\tilde{p})t_j})$ を計算し，U に返す．ただし，(c_p, d_p) は cid に対応する暗号文である．
3. U は $(c_\beta, d_\beta) = (\prod_{i=1}^{k'} c_{\beta_i}, \prod_{i=1}^{k'} d_{\beta_i}) = (g_3^{(r_p - r_{\tilde{p}}) \sum_i t_i}, y_3^{(r_p - r_{\tilde{p}}) \sum_i t_i} \cdot g_3^{(p-\tilde{p}) \sum_i t_i})$ を各 S_j に送信する．
4. 各 S_j は，$\alpha = \mathsf{TCR}(\kappa; u_1, u_2)$ $v'^{(j)} = u_1^{(x_1^{(j)} + y_1^{(j)} \alpha)}$.

$u_2^{(x_2^{(j)}+y_2^{(j)}\alpha)} \cdot d_\beta \cdot (c_\beta^{x_3^{(j)}})^{-1}$ を計算し，$v'^{(j)}$ を U に送信する．u_1, u_2 は cid に対応する暗号文である．

5. U は受け取った $v'^{(j)}$ を復元し，以下の式を得る．
$$v' = u_1^{(x_1+y_1\alpha)} \cdot u_2^{(x_2+y_2\alpha)} \cdot d_\beta \cdot (c_\beta^{x_3})^{-1}$$
$$= u_1^{(x_1+y_1\alpha)} \cdot u_2^{(x_2+y_2\alpha)} \cdot g_3^{(p-\tilde{p})\sum t_i}$$
さらに，$H(v') = K' = (K'_0, K'_1)$ を計算し，K'_1 を各サーバ S_j に送信する．この時，$\tilde{p} = p$ の時は，$K' = K$，それ以外の時は K' は乱数である．
6. 各 S_j は，$a_j = \mathsf{MAC}(K'_1, \chi^{(j)})$ が成立する場合，$\chi^{(j)}$ を U に送信する．成立しない場合，不正とみなし，拒否を示す記号 $\perp$ を U に送信する．
7. U は，$\perp$ を受け取った時，復元失敗としてプロトコルを終了する．$\{\chi^{(j)}\}_{i=0}^{k'}$ を受け取った時，$\{\chi^{(j)}\}_{i=0}^{k'}$ から χ を復元し，5. で生成した K'_0 を用いて χ を復号し，その結果を s' として得る．よって，$\tilde{p} = p$ の時は s' は元の秘密 s，それ以外の時は不正なユーザには元の秘密に関係する情報が漏れない．

この方式は，プロトコルに必ず従う攻撃者である semi-honest adversary に対して安全である．プロトコルに必ずしも従わない攻撃者である malicious adversary に対しても安全な方式にするためには，文献 [3] で使用しているゼロ知識証明を使用する必要がある．

4.4. 複数の秘密を扱う PPSS の安全性議論

この節では，提案方式の安全性について議論する．攻撃者が，semi-honest adversary であり，$k-1$ 台までのサーバと結託し，結託サーバの $st_i = (sk^{(i)}, \chi^{(i)}, a_i)$ を入手できる事を想定する．この時，提案方式が安全性定義を満たすかどうかを考える．

提案方式の直感的に安全であることは以下のように説明できる．まず，攻撃者は，共通鍵で暗号化された秘密の暗号文 χ を復号するため，共通鍵を入手することを考える．そのためには，Kurosawa-Desmedt 暗号の KEM の暗号文を復号して，共通鍵を入手する必要があるが，k 個の秘密状態 st_i あるいは正しいパスワードを入力が必要となる．しかし，攻撃者は，$k-1$ 個以下の秘密状態 st_i しか得ることができず，正しいパスワードを得るためには ElGamal 暗号の秘密鍵が必要であるため，$k-1$ 以下の st_0 から正しいパスワードを推測することはできない．さらに，秘密の暗号文 χ を得るためには、k 個以上の秘密の暗号文シェア $\chi^{(i)}$ が必要となる．このことから，$k-1$ 台のサーバと結託する攻撃者が元の秘密を得るために正しいパスワードが必要であると言える．

次に，既存方式 [3] からの大きな変更点に注目する．変更点は，複数の秘密を扱う点と Kurosawa-Desmedt 暗号を組み合わせている点，秘密の暗号文シェアを保持する点であるが，Kurosawa-Desmedt 暗号の CCA 安全性と秘密分散の安全性から，暗号文等を改竄できないため，これらの変更点において不正はできない．

4.1 節のゲームにおいて，$\frac{1}{2}$ 以上の確率で正しい s_0 か s_1 を得ることを考えた時，攻撃者は何かの方法でそれらの暗号文を区別する必要がある．提案方式はパスワードは ElGamal 暗号によって暗号化，秘密は Kurosawa-Desmedt ハイブリッド暗号によって暗号化後，秘密分散によって分散されており，秘密分散とこれらの暗号は，異なる秘密のシェアや暗号文を区別できない安全性を有している．そのため，暗号文から $\frac{1}{2}$ 以上の確率で s_0 と s_1 を識別することは困難である．正当性は，明らかに満たし，頑強性と健全性は既存方式 [3] と同様である．

以上のことから，提案方式は安全性定義を満たし，$k-1$ 台までのサーバと結託する semi-honest adversary である攻撃者に対しては直感的に安全であるといえる．

5. 比較

この節では，安全性と効率性について提案方式と既存方式 [3] の比較を行う．

5.1. 安全性

■攻撃者による不正

提案モデルで，秘密鍵を固定したハイブリッドバージョンの既存方式と提案方式がパスワードに関係なく元の秘密を復元する不正を検討する．この不正は，攻撃者が結託サーバ上の秘密の暗号文を使用し，正しいパスワードを当てずに，その暗号文を復号を試みる不正である．また，攻撃者は $k-1$ 個のサーバと結託し，全てのサーバは，受け取った暗号文と同じ暗号文が既にサーバにある場合，その暗号文を拒否すると仮定する．既存方式については，以下の通り不正を行うことができる．

1. 攻撃者は結託サーバから正規のユーザの暗号文 $\mathsf{Enc}_{pk}^{Elg}(r)$,$\chi$ を入手する．
2. 攻撃者が設定するパスワード p_A を $\mathsf{Enc}_{pk}^{Elg}(g^{p_A})$ として暗号化し，$Enc_{pk}^{Elg}(r)$ から $Enc_{pk}^{Elg}(2r)$ を計算し，$\mathsf{Enc}_{pk}^{Elg}(2r), Enc_{pk}^{Elg}(g^{p_A})$ をサーバに送信する．
3. パスワード p_A を入力として PPSS を実行すると，$2r$ を出力する．攻撃者は $2r$ から $H_k(r)$ を計算し，χ を $H_k(r)$ で復号する．

つまり，既存方式に対しては攻撃者は s を正しいパスワードなしで入手することができる．

一方，提案方式では，攻撃者は KEM の暗号文 u_1, u_2, χ を結託サーバから入手しようと考える．しかし，攻撃者は，χ を入手するためには，k 個以上の $\chi^{(i)}$ を必要であるため，χ を入手できない．一方，攻撃者が暗号化フェイズで通信路を盗聴できるならば，攻撃者は k 個以上の $\chi^{(i)}$ を入手でき，χ を得ることができる．この場合の不正の流れは以下の通りである．

1. 攻撃者は u_1, u_2, χ を結託サーバ，通信路から入手する．
2. 攻撃者は u_1, u_2 から $2u_1, 2u_2$ を計算し，攻撃者が設定するパスワード p_A を $\mathsf{Enc}_{pk}^{Elg}(g^{p_A})$ として暗号化し，これらを各サーバに送信する．
3. 復元フェイズで，攻撃者は，$\alpha' = \mathsf{TCR}(\kappa; 2u_1, 2u_2)$，$2v'^{(j)} = 2u_1^{(x_1^{(j)}+y_1^{(j)}\alpha')} \cdot 2u_2^{(x_2^{(j)}+y_2^{(j)}\alpha')} \cdot d_\beta \cdot (c_\beta^{x_3^{(j)}})^{-1}$ を計算する．

この時，攻撃者は $2v'^{(j)}$ から v' を計算し，$H(v') =$

(K'_0, K'_1) で共通鍵を生成する．そして，攻撃者は，K'_1 をサーバに送信するが，$\alpha' = \mathsf{TCR}(2u_1, 2u_2) \neq \alpha$ となり，$H(v') = K' \neq K$ となるため各サーバは $\perp$ を返す．

次に，攻撃者が結託サーバを使って元の α を求めることを試みることについて考える．この時，攻撃者は結託サーバの α' を元の α に置き換えることが可能だが，結託していないサーバの α' を置き換えることはできない．そのため，攻撃者は有効な v' を得ることができない．したがって，提案方式に対して不正は不可能である．

パスワードの暗号文に対してもこの不正が可能であると考えられるが，Enc^* にユーザがパスワードを知っていることのゼロ知識証明を付加すると，攻撃者はパスワードの暗号文は知っていてもパスワード自体は知らないため，ゼロ知識証明の値を計算することができない．そのため，この場合もこの不正は不可能である．この不正は，IND-CCA に非常に近い構造であり，提案方式は，IND-CCA 安全性に近い安全性を持つと言える．

■方式全体の安全性

方式全体の安全性について着目して比較を行う．Bagherzandi らの方式は安全性が証明されているが，そのハイブリッドバージョンでは，全体の安全性は証明されていない．ランダムオラクルモデルでは，この方式のハイブリッドバージョンも IND-CPA 安全だと考えられるが，この方式が IND-CCA のような強力な安全性を満たすとは考えられない．

一方，提案方式は，パスワードの暗号化は ElGamal 暗号で行っているため，IND-CPA 安全を満たし，データ暗号化部分を含む全体に対しては Kurosawa-Desmedt 暗号で行うため IND-CCA 安全性を満たし．スタンダードモデルで証明可能である．IND-CCA 安全性は IND-CPA 安全性よりも高い安全性であるため，提案方式のほうが高い安全性を持つといえる．また，ランダムオラクルモデルは妥当性に疑問が残るため，スタンダードモデルでの証明の方が安全性がより明らかとなる．

既存方式は，ユーザ側でのみ間違ったパスワードを拒否することができるが，提案方式はサーバ側で拒否することができる．この点に関しても，提案方式は，サーバ側で不正アクセスを制御することができるため，既存の方式と比較して高い安全性であると言える．

5.2. 効率性

秘密鍵を固定しないハイブリッドバージョンの既存方式の分散後と提案方式の初期化・暗号化後にサーバが持つデータサイズ（暗号文のサイズ，公開鍵サイズ，秘密鍵のサイズの合計）を比較する．Bagherzandi らの方式の暗号文は，群 G 上の 4 つの暗号文と秘密 s の暗号文，公開鍵は群 G 上の 4 個の値であり，秘密毎に必要となる．また，秘密鍵は Z_q 上の単一の値であり，秘密毎に必要である．つまり，$|G| = |Z_q| = \lambda$，秘密の個数を m とした時，全体では $m \cdot (9\lambda + |s|)$ である．

一方，提案方式の暗号文は，群 G 上の 4 つの暗号文と秘密 s の暗号文と MAC のタグ a，公開鍵は，群 G 上の 8 個の値である．また，秘密鍵は Z_q 上の 5 つの値であり，秘密の数に関係なく固定の個数である．つまり，全体では $m \cdot (4\lambda + |s| + |a|) + 13\lambda$ である．

暗号文サイズのみに注目した時，$|a|(= 256\text{bit 程度})$ が増えただけで既存の方式より高い安全性となっているといえる．さらに，$|a| = 256\text{bit}$，$\lambda = 2048\text{bit}$ として，複数の秘密をそれぞれの方式でサーバに分散することを考えた時，サーバ上のデータサイズは，m が 3 以上で提案方式のほうが小さい値になる．そのため，複数の秘密を扱うことを考えた場合，既存方式よりも提案方式のほうがデータ効率が良く，安全性も高いといえる．安全性と効率性の比較を以下の表 1 にまとめる．

表 1　効率性と安全性の比較

	全体の安全性	モデル	サーバ上の値のサイズ
PPSS[3]	N/A	RO	$m \cdot (9\lambda + \|s\|)$
提案方式	IND-CCA like	Standard	$m \cdot (4\lambda + \|s\| + \|a\|) + 13\lambda$

λ:2048bit の値, $|s|$:秘密サイズ,$|a|$:MAC 値 (256bit), m:秘密の個数

6. まとめと今後の課題

本研究では，複数の秘密を考慮した PPSS のモデルとそのモデル上で全体がスタンダードモデルの下で安全になることが期待される方式を構成することができた．提案方式は，IND-CCA 安全性を満たすことが見込まれ，秘密の個数が 3 つ以上の場合，既存方式よりもデータサイズが少なくなる．さらに，$n = 5, k = 3$ で Java で簡易的な実装を行った結果，1 から 10MB までの大きさのファイルに対しては，暗号化に約 500 から 2000ms，復元に約 500 から 1400ms となり，現実的な時間で動作する方式であることが確認できた．この実行時間は通常の秘密分散や既存方式に加えて数百 ms 加わった時間であり，時間増加は少ないといえる．

今後の課題として，提案モデルのもとで提案方式の正確な数学的安全性証明を行うことが挙げられる．

参考文献

[1] 新井貴大, 尾花賢, “Kurosawa-Desmedt 暗号を用いたパスワード認証付き秘密分散法”, 電子情報通信学会 2015 年総合大会, A-7-2, 2015.

[2] T. Arai and S. Obana, “A Password-Protected Secret Sharing based on Kurosawa-Desmedt Hybrid Encryption”, CANDAR’16 WICS, pp.597-603, 2016.

[3] A. Bagherzandi,S. Jarecki,N. Saxena, and Y. Lu, “Password-Protected Secret Sharing”, ACM Conference on Computer and Communication Security, CCS2011, pp.433-444, 2011.

[4] T. ElGamal, “A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”, IEEE Transactions on Information Theory 31 (4), pp.469-472,1985

[5] K. Kurosawa and Y. Desmedt, “A New Paradigm of Hybrid Encryption Scheme”, CRYPTO 2004, LNCS 3152, pp.426-442, 2004.

[6] A. Shamir, “How to Share a Secret”, Comm. ACM 22, pp.612-613, 1979.

[7] V. Shoup, “A Proposal for an ISO Standard for Public Key Encryption (version 2.1)”, manuscript, 2001.